

基于 Cortex_A8 的智能云家居 Web 控制系统设计

孙泽鸿,关维国,刘志建

(辽宁工业大学 电子与信息工程学院,辽宁 锦州 121001)

摘 要: 针对目前智能家居成本昂贵以及用户访问不够灵活的问题,将 Arm 控制器与云端服务器相结合,融入新兴的云计算技术,在 Cortex_A8 处理器上构建嵌入式 Web 服务器,设计了一种新型的智能家居控制系统. 该系统基于 Cortex_A8 和嵌入式 Linux 操作系统,以 Web 编程(CGI)为核心技术,采用 B/S 架构模式,通过 HTML 调用 CGI 应用程序,用户能够在 PC 端或移动终端通过 Web 浏览器与嵌入式硬件进行交互,实现对智能家居设备状态的访问与控制. 测试表明,该系统数据响应及时,运行稳定,用户访问灵活,能够以低成本满足智能家居的应用需求.

关键词: 智能家居;Web 服务器;CGI 编程;云服务器;Linux 操作系统;B/S 架构

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2016)07-0159-05

Design of Intelligent Cloud Home Web Control System Based on Cortex_A8

SUN Ze-hong, GUAN Wei-guo, LIU Zhi-jian

(College of Electronic and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China)

Abstract: Focused on the issue that intelligent home is too expensive and poor user experience, Arm processor was combined with cloud server, joining the emerging cloud computing technology, building embedded Web server on CortexA8 processor, researching and designing a new type of smart home control system. The system was based on Cortex_A8 and embedded Linux operating system, using Web programming (CGI) as core technology, using B/S architecture model as the design structure and call CGI application through HTML. Users could interact with embedded hardware through Web browser in the PC or the mobile phone side, implementing the access and control intelligent household equipment status. The experiment result showed that: the data responds in time, the system runs stably, user accesses flexibility and it meets the demand for intelligent household with low cost.

Key words: smart home; Web server; CGI programming; cloud server; linux operating system; B/S architecture

1 引言

智能家居^[1]不仅给用户提供了安全、便捷的生活环境,还能让用户远程查询和操控家电设备状态. 由于嵌入式设备具有成本低、配置灵活、可靠性高的优点,同时嵌入式操作系统可进行多任务最优调配,因而可确保整个系统运行速度和稳定性.

传统的 Web 智能家居采用的是 C/S 架构开发模式,这种情况下,由于用户使用的操作系统不同,

在对系统软件进行优化时,要进行多个版本的开发,不仅增加了系统优化的经济负担,也降低了系统优化的效率^[2]. 针对这一方面,本设计采用的是 B/S 架构模式的嵌入式系统作为 Web 服务器,成本较低,配置灵活,在提高智能家居系统优化效率的同时也大大降低了成本. 另外,在数据通信和存储方面,本设计在基于物联网的传统智能家居基础上融合了新兴的云计算技术^[2]. 充分利用物联网与云计算的优势,智能家居嵌入式服务器可与云服务器通信并

收稿日期: 2015-09-10; **修回日期:** 2015-10-29

基金项目: 辽宁省博士启动基金资助项目(20131045); 辽宁省教育厅科学研究资助项目(L2012218)

保存用户数据至云端,同时客户端也能从云端访问用户数据,确保数据的安全性和访问的便捷性.

2 智能家居控制系统设计

2.1 系统结构设计

整个系统结构采用了 B/S 模式,各个组成部分之间的关系如图 1 所示. 其中智能家居嵌入式服务器可通过串口及 zigbee 实现对各种家电的集中控制;云服务器实现数据的管理、存储和传输;手机电脑等设备可通过 Web 浏览器,利用网络登录到云服务器,用以实现远程控制家电设备.

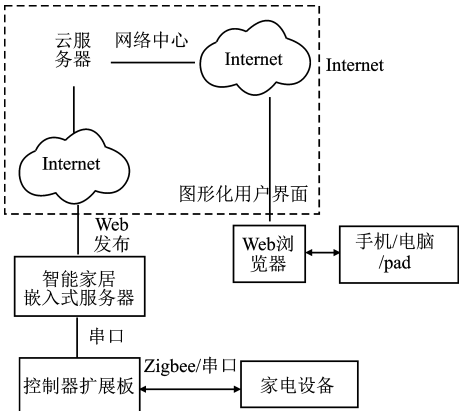


图 1 系统结构图

2.2 硬件平台设计

整个智能家居硬件平台包括核心处理器、人体红外感应模块、距离红外感应模块、烟雾探测器、温湿度检测模块、电机驱动模块、稳压电源模块、液晶显示屏模块、数字时钟模块. 其中人体红外感应模块采用 HC-SR501;距离红外模块采用 IR6721C 实现对空调的控制;电机驱动模块采用 L298N 实现电机的驱动;温湿度检测模块采用器件 DHT11,DHT11 需外接 3.3~5 V 电源,输出数据为串行数据;烟雾探测器采用 MQ-2;液晶模块采用 LCD1602 进行实时显示;电源模块采用了线性 LM7805 和开关稳压芯片 LM2576,给整个系统提供+5 V 的电源. 硬件平台结构如图 2 所示.

为了方便系统的扩展与升级,本系统选择三星公司推出的基于 Arm 体系结构的 Cortex_A8 处理器作为主控制平台,Cortex_A8 采用了 ARMV7 精简指令集,主频可达 1 GHz,实现 2000DIMPS 的高性能运算. 采用 64/32 位内部总线结构,64/32 kB 一级缓存,512 kB 的二级缓存. Cortex_A8 处理器的精心改良设计使得它特别适合应用于性能高、功耗少且成本低的嵌入式设备. 控制器各接口设备参

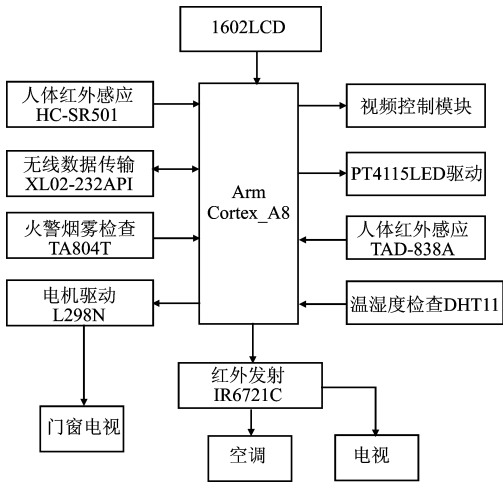


图 2 硬件平台结构图

数如表 1 所示.

表 1 接口设备参数

特性	参数
集成 USB2. 0Hub	拓展出 2 个 USB2. 0 Host 口
串口	4 路,1 路已经转成 RS232,其他预留
中断式按键	8 个,分别定义不同的功能
RJ45 网口	网卡用 DM9000AEP,10/100M 自适应
VGA 输出	1 600×1 200 @100 Hz
USB Slave	1 路,符合 USB2. 0 规范
红外接收头	1 个
ADC 输入	AIN0 连接了在板可掉电阻,用于测试
蜂鸣器	1 路,由 PWM0 控制
EEPROM	AT24C08(256 Byte),用于 I2C 测试
LCD 接口	2 个 LCD 接口,均为 40 Pin 0. 5 mm 间距贴片
SDIO 接口	用于连接 SD,WiFi 模块等
RTC 时钟	带备份电池
PCB	2 层,180×130 mm,方便固定 LCD 模块
供电	DC5V

3 嵌入式 Web 服务器的搭建

3.1 Web 服务器软件构成

本设计的系统核心部分即 Web 服务器,它作为双向通信、远程控制的核心介质,同时也是连接客户端与家居设备、按协议转换数据信息的关键部分. 通过 Web 服务器能够达成室内外网络连接的目标. Web 服务器由 Boa, CGI, SQLite 三部分组成^[3]. Web 服务器以一个小型开源软件的形式运行在类 UNIX 系统中. 其主要功能有以下三方面. 第一,与

客户端建立密切连接,接收客户端发送过来的 HTTP 响应消息;第二,将 HTTP 请求消息实时反馈到客户端;第三,断开连接.作为 Web 应用程序的 CGI (Common Gateway Interface)^[4] 公共网关接口,则提供了 CGI 应用程序进程与 Web 服务器相互通信的标准,是呈现客户端与 Web 服务器之间动态交互的主要形式;CGI 应用程序可以通过网关应用程序中内嵌的 SQL 语句,实现对其数据库有效的访问与操作.作为嵌入式服务器的 SQLite,它使应用程序处于同一个服务器平台.

3.2 构建嵌入式 Web 服务器

为构建 Web 服务器,主机系统采用 VMWare-Ubuntu10.04,编译器采用 arm-linux-gcc-4.1.2,将官方 Boa 源码进行解压并进入到源码目录里面的 src 子目录:

```
$ sudo tar -vxf boa-0.94.13.tar
$ cd boa-0.94.13/src
```

输入命令,生成 Makefile 配置文件:

```
$ sudo ./configure
```

为使其能在 linux 系统中运行,需要修改 Makefile 配置文件里面的交叉编译工具,使用命令 make 对其进行编译,结果可得到执行程序 boa,将其调试信息滤除,最后得到大约 60 kB 大小的程序.

```
$ make
```

make 成功后,可使用 file 命令查看生成的 boa 可执行文件.

```
$ file boa
```

ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.27, not stripped

最后在 boa.config 文件里面配置 Boa,使其能够支持 CGI 程序的执行,拷贝 boa 可执行文件到根文件系统的/bin 目录下,即可运行 Boa.

```
# /bin/boa
```

开发流程如图 3 所示.

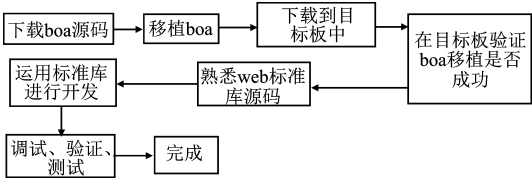


图 3 嵌入式 Web 服务器开发流程

Cgic 库的移植过程类似 Boa 的移植,下载编译成功之后,在宿主机浏览器输入相应的地址,http://192.168.1.3/cgi-bin/cgicest.cgi,从图 3

可以看出,测试脚本和 Cgic 库已经成功完成移植.测试结果如图 4 所示.

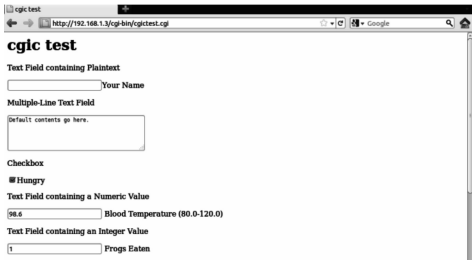


图 4 Cgic 库移植测试结果

4 云端数据通信的设计

由于云服务器独有的优势,在智能家居中,云服务器承担三项工作:接收核心处理器送来的传感器信息数据,并对其进行解析、保存;根据预先设定,适时将控制命令通过网络发送给核心处理器,实现实时控制、定时控制、智能控制;提供网页交互界面,供用户查看家居状态,控制相应设备.

4.1 云端与手机端数据通信处理模块

此模块的功能主要完成云服务器接收到的网络数据的分析、数据的转发(云端到手机客户端与手机客户端到云端)、数据库的数据同步操作(数据库的数据保存、添加、查询、删除等)、实现操作结果的状态反馈.数据来源于网络通信模块接收到的请求数据.基本数据结构与部分函数声明如下:

```
/* 拆解云端数据并且包装成手机端要的形式函数 */
void B2D_process(char * chat,int macid)
/* 异常判断函数 异常则调用日志接口 */
void abnormal_judge(int macid);
/* 处理成手机端要的格式 */
void turn2D();
void depart_threshold(char * DcmdorTshd_buf) //拆解
解阈值数据
void change_threshold()//更新阈值数据
void turn2B() //将数据转换成云端要
的格式
void depart_cmd(char * DcmdorTshd_buf) //拆解命令
的函数
void D2B_process(char * chat,int macid) //拆解手机
端数据并且包装成云端要的形式函数
void depart_usr_info() //拆解用户注册登录时数据
的函数
```

手机端通过 Socket 与云端建立 TCP 连接,云端的链表将保存手机用户的 IP 地址,以此双向通信.由于手机端与云端数据处理格式不同,云端服务器必须有一个数据转换模块,进行数据格式转换.数

数据库模块用以存放和转发手机端或者嵌入式服务器所发送的请求数据. 云端与手机端的网络通信方式如图 5 所示.

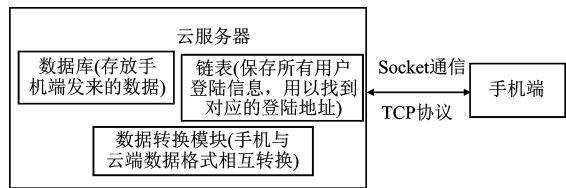


图 5 云端与手机端通信方式示意图

4.2 网络通信模块

网络通信模块的功能主要实现云服务器网络端口建立、云服务器数据的接收和云服务器数据的发送. 数据来源于网络通信模块接收到的客户端的传输数据. 基本数据结构与部分函数声明如下:

```
void tidy(conect_usr_info * user_usr_info) //整理连接上来的用户

int main(int argc,char * argv[]) //服务器,负责接受和转发
```

5 测试与分析

5.1 串口通信的实现方案

在智能家居控制板与家电设备的串口通讯中, 确定一个明确而合理的通讯协议是关键. 首先必须对串口参数进行配置, 主要包括波特率、数据位、停止位、流控协议. 对串口参数的配置可通过以下结构体实现:

```
struct termios
{
    flag_t c_iflag; //input flags
    flag_t c_oflag; //output flags
    flag_t c_cflag; //control flags
    flag_t c_lflag; //local flags
    cc_t c_cc[NCCS]; //control characters
};
```

其次, 为了在程序中能方便表示出各个关键字, 必须明确规定信息家电各个状态的功能码, 在程序中采用十六进制数据来表示串口帧的每个字段, 串口帧协议的具体定义及参数说明如表 2 所示.

5.2 CGI 控制程序设计

Cgic 库移植成功之后, 进行程序设计^[5]. 程序设计流程如图 6 所示. 在本设计中, CGI 程序分为三个部分: 登陆及鉴权、控制部分以及查询与显示.

用户登陆账号和登录密码存放在日志里面, 用户登录的时候, 调用一个 CGI 程序, 将用户输入的

信息与存放账户和密码的日志文档做一个对比, 相同, 则登录成功, 进入主界面. 否则登录失败, 重新输入.

表 2 串口协议功能码及参数说明

数据名称	数据类型	(参数)说明
前导符	0x5354	一帧的开始
终端地址	0xe0	智能终端地址
目的地址	0x37	串口设备地址
电灯	0x11	电灯
温湿度传感器	0x12	温度误差:±2℃ 湿度误差:5%
直流电机	0x13	模拟窗帘
步进电机	0x14	模拟空调
烟雾传感器	0x15	检测浓度:300~1 000 ppm
距离红外传感器	0x16	探测距离:8 m
人体红外传感器	0x17	延时时间:0.5~20 s
智能开关(开)	0x06	开状态
智能开关(关)	0x00	关状态

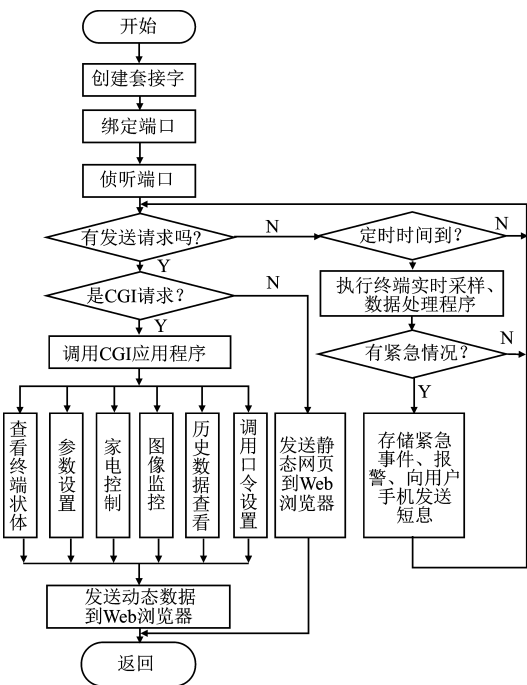


图 6 程序设计流程图

控制部分主要完成对各种家电的开关控制. 以控制房间灯为例, 在安装房间灯驱动的前提下, 通过函数 led1. get_led_status(int) 获取此时灯的状态, 通过函数 led_set(int, int) 控制灯的状态, 通过函数 getenv("QUERY_STRING") 获取开关页面传送过

来的按钮名称,按钮点击一次,状态将改变一次。

显示部分是对各种传感器进行读值操作^[6]。安装驱动之后,通过函数 tmp_hum. check(&tmp, &hum) 获取房间的实时温度与湿度,并通过函数 temp_Max. get_value(&temp_pck) 设置报警温度值,温度超过此设置值,蜂鸣器将响应。

部分 CGI 程序代码如下:

```
int main(int argc, char * * argv) // 灯
{
    int a, b, c, d;
    a = ! (led1. get_led_status(1));
    b = ! (led1. get_led_status(2));
    c = ! (led1. get_led_status(3));
    d = ! (led1. get_led_status(4));
    // 获取各号灯状态信息, 1 为亮, 0 为暗
    return(0);
}

int check_temperature_humidity(unsigned short int * t,
unsigned short int * h) // 温湿度
{
    int fd;
    unsigned long temperature = 0;
    fd = open("/dev/humidity", 0);
    read(fd, &temperature, sizeof(temperature));
    * h = (temperature & 0xff000000) >> 24;
    * t = (temperature & 0x0000ff00) >> 8;
    close(fd);
    return 0;
}

int infraerd_setstat(char * filename, char stat) // 红外
{
    int fd, fd1;
    char buf[2] = "\0";
    fd = open(filename, O_RDWR);
    if(fd < 0)
        printf("error\n");
    if((fd1 = open("/dev/infrared", O_RDWR)) < 0)
    {
        perror("open:");
        return -1;
    }
    read(fd1, &buf, sizeof(buf));
    if(lock_unlock(fd, F_RDLCK) < 0)
        return -1;
    write(fd, buf, 1);
    lock_unlock(fd, F_ULOCK);
    close(fd);
```

```
close(fd1);
```

```
return 0;
```

```
}
```

5.3 程序编译与实验结果

程序设计完成,对其进行交叉编译,将程序编译成开发板能执行的文件。在编译无误之后,把所有的. cgi 文件拷贝至根文件系统的/www/cgi-bin/目录下,使用语句 cp *. cgi /opt/rootfs/cgi-bin 即能完成拷贝。并将相对应的 Html 文件以及其他相关文件拷贝到指定文件夹里面^[10]。

实验过程中,将目标板与电脑分别通过网线接入以太网,并设置 PC 端的 IP 地址与核心板的 IP 地址在同一个网段上。接着在 PC 机打开浏览器,输入目标板的 IP 地址 http://10. 9. 12. 221,在登陆界面相应位置输入正确的用户名与密码,便可登录到主控界面。

接下来对目标板上所模拟的家用电器进行控制,成功完成了 Web 网页与底层硬件的交互。另外也可利用其他智能终端。至此,完整完成了智能家居控制系统的设计,其后续功能可进一步补充,拓展和开发。

6 结束语

本文所设计的智能家居控制系统以 CGI 原理为基础,以云服务器数据库为后台,在 Arm Cortex_A8 处理器上构建嵌入式 Web 服务器,通过互联网实现外部对该系统的连接与控制。使用 HTML 与 C 语言编程,实现了用户、Web 服务器以及网关应用程序之间的动态交互。实验结果表明,本文所设计的系统具有实用性,相对于传统智能家居系统,提升了用户访问的灵活性与实时性,降低了成本。在后续的研究工作中,将着重进行 android 移动客户端 app 的开发以及基于语音识别的智能家居控制研究。

参考文献:

- [1] Feng Z R, Wang H M. Embedded wireless smart home monitoring system[J]. Electronic Design Engineering, 2015, 23(5):1758-1761.
- [2] 李海涛. 云计算用户数据传输与存储安全研究[J]. 现代电子技术, 2013, 6(1):24-26.
- [3] 毛明毅,蒋元恒,陈志成. 智能家居远程 Web 管理控制平台的设计与实现[J]. 微电子学与计算机, 2013, 30(5):121-124.
- [4] 李晓丹. 基于 STM32 的物联网嵌入式网关的设计[J]. 计算机工程与应用, 2015, 12(4):61-65.

(下转第 168 页)

动器在每次将事务送出后,通过 callback 回调 Coverage,进行覆盖率统计.

经过大量的激励测试,得到覆盖率报告,报告如图 6 所示.从报告中可以看到,cover_arg 覆盖率组中的各个变量覆盖点以及交叉覆盖点的覆盖率均达到了 100%,最终总覆盖率为 100%,达到了期望的覆盖率.这说明测试点已经得到了充分的验证^[6].在验证过程中,如果发现还未被覆盖的地方,可以通过增加特定的测试激励,实现覆盖,直到最终覆盖率为 100%为止.

Questa Covergroup Coverage Report

Scope: /my_pkg/my_coverage

Show All Show Covered Show Missing

Covergroups / Instances	Total Bins	Covered Bins	Goal	Coverage	% of Goal
Covergroup cover_arg	345	345	100.00%	100.00%	100.00%
Instance /my_pkg/my_coverage/cover_arg	345	345	100.00%	100.00%	100.00%

Covergroup type: cover_arg	100.00%	100.00%	100.00%
Coverpoints / Bins	At Least	Hits	Goal
Coverpoint: cov_r_del			100.00%
Coverpoint: cov_r_tari			100.00%
Coverpoint: cov_r_pw			100.00%
Coverpoint: cov_r_RTcal			100.00%
Coverpoint: cov_r_TReal			100.00%
Coverpoint: cov_r_x			100.00%
Coverpoint: cov_r_TReal			100.00%
Coverpoint: cov_f_cg			100.00%
Coverpoint: cov_b_cnt			100.00%
Cross: cov_tari_pw			100.00%
<cov_r_tari,cov_r_pw>			100.00%
Cross: cov_tari_RTcal			100.00%
<cov_r_tari,cov_r_RTcal>			100.00%
Cross: cov_tari_x			100.00%
<cov_r_tari,cov_r_x>			100.00%
Cross: cov_RTcal_TReal			100.00%
<cov_r_RTcal,cov_r_TReal>			100.00%

图 6 覆盖率报告

5 结束语

通过采用 UVM 验证方法学设计的验证平台,与传统的验证平台相比,对于不同工程具备相同特

征的,可以实现部分复用;受约束的随机激励可以有效覆盖人工难以预期的边界条件;覆盖率分析有助于验证的完备性.这缩短了开发周期,提高了验证效率.验证结果表明,该验证平台的覆盖率满足了出口条件,达到了预期效果,有效提高了验证效率和质量.该验证平台的一些思想,为其他系统级的芯片验证提供了很好借鉴意义.

参考文献:

[1] 罗莉,何鸿君,窦强,等.覆盖率驱动的芯片功能验证设计与实现[J].计算机工程与技术,2013,35(1):36-40.

[2] 田劲,王小力.基于 UVM 验证方法学的 AES 模块级验证[J].微电子学与计算机,2012,29(8):86-90.

[3] Accellera, Universal Verification Methodology(UVM) 1.1 User's Guide[S]. 2011.

[4] EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz-960MHz Version 1.2.0[S]. 2008.

[5] 克里斯 斯皮尔. System Verilog 验证测试平台编写指南[M].北京:科学出版社,2009:78-87.

[6] 朱车壮,陈岚.基于覆盖率驱动的 SoC 验证技术研究[J].微电子学与计算机,2011,28(11):48-51.

作者简介:

王国军 男,(1989-),硕士研究生.研究方向为大规模数字集成电路设计与测试. E-mail:wgjntu@163.com.

景为平 男,(1954-),研究员,博士生导师.研究方向为集成电路设计及其在智能交通领域的应用.

(上接第 163 页)

[5] 罗正军,罗鹏辉,王晓娟,等.基于 Linux 系统和 ZigBee 的智能家居系统[J].现代电子技术,2012,35(1):37-39.

[6] 黄媛.无线传感器网络中一种基于可靠性的数据收集算法[J].计算机工程,2015,41(2):85-90.

作者简介:

孙泽鸿 男,(1990-),硕士研究生.研究方向为通信信号处理. E-mail:912405172@qq.com.

关维国 男,(1973-),副教授,博士,硕士生导师.研究方向为移动网络定位、通信信号处理.

刘志建 男,(1990-),硕士研究生.研究方向为移动网络定位.